UNLIMITED

2

# RSRE
# MEMORANDUM No. 4131

# ROYAL SIGNALS & RADAR ESTABLISHMENT



## COMPARISON OF INTERPOLATION ALGORITHMS FOR SPEED CONTROL IN AIR TRAFFIC MANAGEMENT

Author: A J Budd



PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
R S R E MALVERN,
WORCS.





UNLIMITED

CONDITIONS OF RELEASE

0032583 BR-109213

****************************************** U



****************************************** Y

Reports quoted are not necessarily available to members of the public or to commercial organisations.

**ROYAL SIGNALS AND RADAR ESTABLISHMENT**

**Memorandum 4131**

TITLE: COMPARISON OF INTERPOLATION ALGORITHMS FOR SPEED CONTROL IN AIR TRAFFIC MANAGEMENT

AUTHOR: A J BUDD

DATE: August 17, 1988

**Abstract**

With air traffic movements at a high level, techniques to assist air traffic management using computers are being investigated. One technique in particular being studied is the early adjustment of the speed of arriving aircraft so that the rate of flow near to the airports is closely matched to landing capacity.

A Speed Control Adviser has been developed which allocates a landing time to each inbound aircraft. Once the estimated landing time is known, the speed the aircraft must fly needs to be calculated. This cannot be done directly and interpolation using a suitable polynomial approximation is used.

This memorandum investigates four polynomials and examines their effectiveness at providing a good estimate with minimum computation.

CARDPB Ref : 2.2.1.2
RSRE Task : 636
CAA Sponsors : D PE(LD)
CAA Liason Officer : D Dawson

## Contents

# 1 Introduction

Air traffic movements in the London area are at a high level and are expected to increase. Techniques involving the use of computer assistance are being investigated to enable air traffic management and control systems to handle this increase. One technique in particular being studied by the Terminal Control Systems Development Group (TCSDG) in AD4 Division of RSRE, is the early adjustment of the speed of arriving aircraft so that the rate of flow near to the airports is closely matched to landing capacity [1].

The TCSDG is developing a Speed Control Advisor (SCA) [2,3]. This uses an algorithm which includes the allocation of a landing slot to each inbound aircraft based on a specified runway separation. The calibrated airspeed (CAS) required for descent in order that the slot time is achieved needs to be computed. That is, given the time over which an aircraft must fly from A to B (usually descent) at what speed must it travel?

The speed value needed to cause an aircraft to take a specified time to fly from some point A to another point B along a specified route through a specified wind field cannot be calculated directly. The reasons for this are that aircraft normally operate at constant CAS or constant Mach Number rather than at constant true air speed (TAS), that different aircraft types perform differently especially in descent and turns, and that wind speed and direction can vary rapidly with change of altitude. However, the time taken to fly from A to B at a specified CAS can be calculated, and by doing this calculation for several speed values and using some form of interpolation, the required speed can be found.

The time to fly from A to B can be expressed as

$$T = \int_A^B \frac{ds}{|\vec{V_g}|}$$

where

$$\vec{V_g} = \vec{V_t} + \vec{W}$$

$\vec{V_g}$ is the ground velocity vector, $\vec{V_t}$ is the true velocity vector and $\vec{W}$ is the wind vector. $\vec{V_t}$ is assumed to head in a direction such that $\vec{V_g}$ points along the route. $\vec{V_t}$ is a function of CAS and altitude, or of Mach number and altitude, depending on the phase of flight.

A set of procedures for performing these time calculations was written for use in the TCSDG real-time system, and was available for use in this investigation.

For each aircraft, the SCA finds the time taken to descend at maximum speed, minimum speed and preferred speed (the speed at which the aircraft would like to descend). These known points are substituted into a polynomial approximation to give an estimate (chosen speed) for the required CAS (target speed). The time to fly the route portion at the chosen speed is calculated using the TCSDG procedures, and is compared to the target time. If the difference is not within acceptable limits then further iterations are needed. As each calculation of time is lengthy it is important to find a polynomial that gives a good estimate quickly.

This note investigates four approximations that could be used and examines their effectiveness at providing a good estimate with minimum computation.

## 2 The Approximations

For an aircraft flying a typical descent profile, the relationship between speed and time is represented by the actual data curve in Figs(1)-(4),(10)-(12). (See section 4 for a detailed description of descent profile used).

Four approximations to this curve have been tried. In the derivations that follow the notation used is

$$\begin{array}{lllllll} I_f & = & maxcas & T_f & = & \textit{time to fly descent at maxcas} \\ I_s & = & mincas & T_s & = & \textit{time to fly descent at mincas} \\ I_p & = & prefcas & T_p & = & \textit{time to fly descent at prefcas} \\ I_c & = & \textit{chosen speed} & T_c & = & \textit{time to fly descent at chosen speed} \\ I_t & = & \textit{target speed} & T_t & = & \textit{target time} \end{array}$$

where $I_s < I_p < I_f$ and $T_f < T_p < T_s$ (since it takes longer to fly at a slower speed).

### 2.1 Approximation I

Initially, only two known points were used ie $(I_f, T_f)(I_s, T_s)$ in finding a suitable approximation. The first assumption made was that the relationship between speed and time was inversely proportional

$$I = k/t \qquad (k\ const)$$

We can find some constant $k$ such that

$$\begin{aligned} I_f - I_s &= k(1/T_f - 1/T_s) \\ &= k(T_s - T_f)/(T_f T_s) \end{aligned}$$

$$\Rightarrow k = (I_f - I_s)T_f T_s/(T_s - T_f) \tag{1}$$

Also, given that $I_c$ is proportional to $T_c$ then

$$\begin{aligned} I_c - I_f &= k(1/T_t - 1/T_f) \\ &= k(T_f - T_t)/(T_f T_t) \end{aligned}$$

$$\Rightarrow I_c = I_f + k(T_f - T_t)/(T_f T_t)$$

Substituting for $k$ from (1) gives

$$\begin{aligned} I_c &= I_f + (I_f - I_s)(T_f - T_t)T_f T_s/T_f T_t(T_s - T_f) \\ &= I_f + (I_f - I_s)(T_f - T_t)T_s/T_t(T_s - T_f) \end{aligned}$$

Fig(1) shows the curve this equation represents plotted against the data being modelled.

The assumption that $I = k/t$ appears to provide a reasonable model and this approximation was tested along with the others to see what sort of estimate would be produced.

The use of only two points is not ideal. It is more likely that a better fit will be obtained with more points and as three are being calculated already (ie preferred speed and time), three should be used.

## 2.2 Approximation II

Approximation II is an extension of approximation I which makes use of the three points. Speed is still assumed to be inversely proportional to time but as well as the curve shaping constant $k$, two further constants, $a$ and $b$, are included which allow the curve to be moved to fit the data.

$$I + a = \frac{k}{(t + b)} \tag{2}$$

We have three points $(I_f, T_f)(I_p, T_p)$ and $(I_s, T_s)$ which when substituted into (2) give the equations

$$\begin{aligned} I_f + a &= k/(T_f + b) \\ I_p + a &= k/(T_p + b) \\ I_s + a &= k/(T_s + b) \end{aligned}$$

These equations can be solved to give the following values for $b$ , $a$ and $k$.

$$\begin{aligned} b &= \frac{I_f T_f (T_s - T_p) + I_p T_p (T_f - T_s) + I_s T_s (T_p - T_f)}{I_f (T_p - T_s) + I_p (T_s - T_f) + I_s (T_f - T_p)} \\ a &= \frac{I_p (b + T_p) - I_f (b + T_f)}{(T_f - T_s)} \\ k &= (I_f + a)(T_f + b) \end{aligned}$$

which are substituted into (2) to give an approximation of speed given time. The three points $(I_f, T_f)(I_p, T_p)$ and $(I_s, T_s)$ must be distinct in order to provide three solvable simultaneous equations.

Fig(2) shows the curve this equation represents plotted against the data being modelled.

## 2.3 Approximation III

A possible numerical analysis solution to the problem of fitting a function to some data points is to use the Lagrange polynomial [4,5]. This is a polynomial that can be constructed in such a way that it passes through each known data point and is of degree less than or equal to the number of points provided minus one. For example if we know three points, the Lagrange polynomial which passes through these points will be quadratic or linear.

### 2.3.1 Construction of the Lagrange polynomial

Assume that, given $n+1$ distinct points $t_i$ $(i = 0, 1, \ldots, n)$ on the interval $\Psi$ with function values $f(t_i) = f_i$ $(i = 0, 1, \ldots, n)$ we can construct a polynomial of degree $\leq n$ which passes through the $n + 1$ points.

The Lagrange method is to begin by expressing this polynomial as

$$I(t) = L_0(t) f_0 + L_1(t) f_1 + \ldots + L_n(t) f_t$$

where each $L_k(t)$ $(k = 0, 1, \ldots, n)$ is a polynomial of degree not exceeding $n$. In order for the polynomial to pass through the $n + 1$ points given, the following condition must hold

$$I(t_i) = f_i \quad (i = 0, 1, \ldots, n)$$

ie. at any of the given points, the Lagrange polynomial value is equal to the actual function value. This condition holds if

$$L_k(t_i) = \begin{cases} 1 & k = i \\ 0 & k \neq i \end{cases}$$

So for example

$$\begin{aligned} I(t_1) &= L_0(t_1)f_0 + L_1(t_1)f_1 + \ldots + L_n(t_1)f_n \\ &= 0 * f_0 + 1 * f_1 + 0 * f_2 + \ldots + 0 * f_n \\ &= f_1 \end{aligned}$$

The family of polynomials which satisfy the requirements above are of the form

$$L_k(t) = \frac{(t - t_0)(t - t_1)\ldots(t - t_{k-1})(t - t_{k+1})\ldots(t - t_n)}{(t_k - t_0)(t_k - t_1)\ldots(t_k - t_{k-1})(t_k - t_{k+1})\ldots(t_k - t_n)}$$

Note that the numerator of $L_k(t)$ is a product of all factors of the form $(t - t_i)$ except $(t - t_k)$. Since each $L_k(t)$ is a polynomial of degree $n$ and since

$$I(t) = L_0(t)f_0 + L_1(t)f_1 + \ldots + L_n(t)f_n$$

then $I(t)$ will be a polynomial of degree $\leq n$.

### 2.3.2 Use of the Lagrange polynomial

We have three known points $(I_f, T_f)(I_p, T_p)(Is, T_s)$ so $n = 2$ and the Lagrange polynomial can be constructed as follows

$$I(t) = L_0(t)I_f + L_1(t)I_p + L_n(t)I_s$$

where

$$L_0(t) = \frac{(t-T_p)(t-T_s)}{(T_f-T_p)(T_f-T_s)} \quad L_1(t) = \frac{(t-T_f)(t-T_s)}{(T_p-T_f)(T_p-T_s)}$$

$$L_2(t) = \frac{(t - T_f)(t - T_p)}{(T_s - T_f)(T_s - T_p)}$$

Fig(3) shows the curve this equation represents plotted against the data being modelled. It is clearly a reasonable fit and should provide a good estimate of the speed given time.

## 2.4 Approximation IV

The fourth approximation assumes that the speed/time relationship can be represented by an inverse cubic.

$$I(t) = \frac{a_0}{t} + \frac{a_1}{t^2} + \frac{a_2}{t^3} \tag{3}$$

We have three points $(I_f, T_f)(I_p, T_p)$ and $(I_s, T_s)$ which when substituted into the inverse cubic give the equations

$$I_f = a_0/T_f + a_1/T_f^2 + a_2/T_f^3 \tag{4}$$

$$I_p = a_0/T_p + a_1/T_p^2 + a_2/T_p^3 \tag{5}$$

$$I_s = a_0/T_s + a_1/T_s^2 + a_2/T_s^3 \tag{6}$$

These can be solved to give values for $a_0$ , $a_1$ and $a_2$ which are substituted into (3) to give an approximation for speed given time. The three points $(I_f, T_f)(I_p, T_p)$ and $(I_s, T_s)$ must be distinct in order to provide three solvable simultaneous equations.

### 2.4.1 Calculation of $a_0$ , $a_1$ , $a_2$

We can write eqns (4)-(6) as

$$\begin{pmatrix} I_f \\ I_p \\ I_s \end{pmatrix} = \begin{pmatrix} 1/T_f & 1/T_f^2 & 1/T_f^3 \\ 1/T_p & 1/T_p^2 & 1/T_p^3 \\ 1/T_s & 1/T_s^2 & 1/T_s^3 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}$$

represented by

$$\bar{I} = T\bar{a}$$

and so if $T^I$ is the inverse of $T$ then

$$T^I\bar{I} = T^I T\bar{a} = \bar{a}$$

The inverse of $T$ is defined as

$$T^I = adj\ T/\ det\ T$$

where $adj\ T$ is the transpose of the matrix of cofactors of $T$ and $det\ T$ is the determinant of $T$. As long as $T_f \neq T_p \neq T_s$ then $det\ T$ will not equal zero. From this we can find $a_0$ , $a_1$ and $a_2$

Let $B$ be the matrix of cofactors of $T$. Then

$$B = \begin{pmatrix} \frac{1}{T_p^2T_s^3} - \frac{1}{T_p^3T_s^2} & -\frac{1}{T_pT_s^3} + \frac{1}{T_p^3T_s} & \frac{1}{T_pT_s^2} - \frac{1}{T_p^2T_s} \\ -\frac{1}{T_f^2T_s^3} + \frac{1}{T_f^3T_s^2} & \frac{1}{T_fT_s^3} - \frac{1}{T_f^3T_s} & -\frac{1}{T_fT_s^2} + \frac{1}{T_f^2T_s} \\ \frac{1}{T_f^2T_p^3} - \frac{1}{T_f^3T_p^2} & -\frac{1}{T_fT_p^3} + \frac{1}{T_f^3T_p} & \frac{1}{T_fT_p^2} - \frac{1}{T_f^2T_p} \end{pmatrix}$$

$$= \frac{1}{T_f^3T_p^3T_s^3} \begin{pmatrix} T_f^3(T_p - T_s) & T_f^3(T_s^2 - T_p^2) & T_f^3(T_sT_p^2 - T_s^2T_p) \\ T_p^3(T_s - T_f) & T_p^3(T_f^2 - T_s^2) & T_p^3(T_fT_s^2 - T_f^2T_s) \\ T_s^3(T_f - T_p) & T_s^3(T_p^2 - T_f^2) & T_s^3(T_pT_f^2 - T_p^2T_f) \end{pmatrix}$$

The $adj\ T$ is the transpose of the matrix of cofactors, that is the transpose of $B$. Hence $adj\ T =$

$$\frac{1}{T_f^3T_p^3T_s^3} \begin{pmatrix} T_f^3(T_p - T_s) & T_p^3(T_s - T_f) & T_s^3(T_f - T_p) \\ T_f^3(T_s^2 - T_p^2) & T_p^3(T_f^2 - T_s^2) & T_s^3(T_p^2 - T_f^2) \\ T_f^3(T_sT_p^2 - T_s^2T_p) & T_p^3(T_fT_s^2 - T_f^2T_s) & T_s^3(T_pT_f^2 - T_p^2T_f) \end{pmatrix}$$

The inverse of $T$ is equal to $adj\ T/det\ T$

$$det\ T = \frac{T_f^3T_p^3T_s^3}{T_f^2(T_p - T_s) + T_p^2(T_s - T_f) + T_s^2(T_f - T_p)}$$

We have

$$T^I\bar{I} = \bar{a}$$

and so

$$a_0 = \frac{I_fT_f^3(T_p - T_s) + I_pT_p^3(T_s - T_f) + I_sT_s^3(T_f - T_p)}{T_f^2(T_p - T_s) + T_p^2(T_s - T_f) + T_s^2(T_f - T_p)}$$

$$a_1 = \frac{I_fT_f^3(T_s^2 - T_p^2) + I_pT_p^3(T_f^2 - T_s^2) + I_sT_s^3(T_p^2 - T_f^2)}{T_f^2(T_p - T_s) + T_p^2(T_s - T_f) + T_s^2(T_f - T_p)}$$

$$a_2 = \frac{I_fT_f^3(T_sT_p^2 - T_s^2T_p) + I_pT_p^3(T_fT_s^2 - T_f^2T_s) + I_sT_s^3(T_pT_f^2 - T_p^2T_f)}{T_f^2(T_p - T_s) + T_p^2(T_s - T_f) + T_s^2(T_f - T_p)}$$

Fig(4) shows the curve represented by the inverse cubic compared to the actual data.

## 3 Running the experiment

The aim of this investigation was to find the best approximation modelling the relationship between speed and time. It is possible to calculate the time taken to fly a descent given the speed by using a series of complex prediction procedures. This process is lengthy so the best approximation will be one that takes minimum calls of these procedures to deliver a reasonable estimate (ie. one within specified limits) of speed given time .

One program for each algorithm was written in CORAL 66. The programs were basically the same but had slight variations according to the algorithm being used. The programs included calls to the prediction procedures which calculate the time to fly the descent given the speed. The route portion chosen was of an aircraft approaching Heathrow from the North descending from a height of 31,000ft to 8,000ft (stack base level). The descent profile consisted of a cruise section at the Mach equivalent of 290k followed by a descent at the chosen speed. This descent was subject to minimum calibrated airspeeds at defined levels illustrated in Fig(5).

Initially, the times taken to fly the route portion at max speed, min speed and preferred speed were found, providing the three points (360, 1153),(210, 1569), (260, 1333). These were used to find the constants in each algorithm. A target time, $T_t$ , was chosen ie. the time the aircraft had to fly the descent and an estimate for the speed required to fly the descent was found by substituting the target time into each approximation. Having obtained this value for the speed, the investigation proceeded to see how reasonable it was.

The prediction procedures were called to calculate the actual time an aircraft would take given this estimated speed. If the difference between actual time and target time was less than 5 secs, the estimated speed was returned as the chosen speed, $I_c$ ,(the speed needed to fly the descent). If the difference was greater than 5 secs, further iterations were required.

### 3.1 Further iterations

The calculation of the time to fly the route portion at estimated speed provided an extra point $(I_c, T_c)$. Since $T_c$ was nearer to the target time than at least one of the end points, it was used in the further iterations to obtain a better estimated speed.

#### 3.1.1 Approximation I

The obvious method to use with this type of approximation was to shorten the interval over which the interpolation was taking place. This was achieved using the secant method.

#### 3.1.2 Approximation II

The method used with this approximation shortened the interval over which interpolation was taking place. In this case the interval was over $I_s$ to $I_f$ with $I_p$ in between. Moving of the end points had to take into account all three. The extra point $(I_c, T_c)$ was taken to be the mid point of the next three points used since a better fit was achieved. The end points were then picked accordingly.

### 3.1.3 Approximation III

The extra point $(I_c, T_c)$ was used in addition to the three original points, to find the next order Lagrange approximation eg. one with degree $\leq 4$. This gave a function that passed through the four known points.

### 3.1.4 Approximation IV

The method used with this inverse cubic approximation was the same as for approximation II.

## 3.2 Output

Each program was run with target times ranging from 1165 to 1548 secs which produced speeds between 214 and 329 knots.

| Target time | 1165 | 1195 | 1236 | 1278 | 1319 | 1361 | 1403 | 1444 | 1486 | 1527 | 1548 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Speed | 329 | 309 | 290 | 276 | 264 | 253 | 244 | 236 | 228 | 219 | 214 |

The target times were chosen to give a fairly even spread over the range between $T_f$ and $T_s$.

The output recorded for each target time was

- estimated speed
- actual time for descent at estimated speed
- final chosen speed and time
- number of iterations required

The algorithms were compared by looking at the number of iterations required to achieve a good estimate. This shows how often the prediction procedures were called giving an indication of how good the algorithm was and how closely it modelled the speed time relationship.

The Extended inverse proportional, the Lagrange and the Inverse cubic approximations were further examined to see how the choice of the third point (initially the preferred cas) affected the reasonableness of the estimate.

# 4 Results

A "reasonable estimate" of the speed was defined as one whose associated time to fly the route portion was within 5secs of the target time. Every computation of an estimated speed, including the first, was considered to be an iteration. The number of iterations required for each algorithm are shown in Fig(6).

From this it can be seen that the Extended inverse proportional, Inverse cubic and Lagrange algorithms provide good approximations, taking at most 2 iterations to give a reasonable estimate. The Inverse proportional algorithm provides a fair approximation, taking 3 iterations at most speeds but less near each end point.

It can be seen that the algorithms using three points provide a better model than the Inverse proportional, especially around the preferred speed mark. This is to be expected since these three polynomial approximations are constructed in such a way that they pass through the three original points $(I_f, T_f)(I_p, T_p)(I_s, T_s)$. The implication of this is that the choice of preferred speed has an effect on the accuracy of the model, the question being "How much ?".

In order to study this effect, the experiment was run with preferred speeds of 230k, 275k and 320k. Figs (7),(8),(9) show the difference between the number of iterations required for each algorithm.

The figures show that for speeds around the preferred speed and two end points, the estimate is close to the target. Between these points, the estimates get worse as the models become less accurate.

With a preferred speed of 230k (Fig(7)), the algorithms are providing reasonable models, taking mostly two iterations although the Inverse cubic and Lagrange take three iterations at speeds just below 340k.

With a preferred speed of 275k (Fig(8)), the actual midway point between the end speeds, all three algorithms provide good approximations taking no more than two iterations.

A preferred speed of 320k (Fig(9)) gives rise to the least accurate estimates. The Lagrange algorithm is particularly affected by this high preferred speed taking three or four iterations to converge on speeds away from the end points. The Inverse cubic and Extended inverse proportional are also affected but not to such a great extent.

These results are illustrated by Figs(10),(11) and (12) which show the graphs of actual data along with the graphs of the Extended inverse proportional, Inverse cubic and Lagrange algorithms with differing preferred speeds. These demonstrate the effectiveness of each algorithm to provide a model of the actual data.

## 5 Conclusions

The four possible relationships considered between speed and time were

1. inverse proportional ie $I = k/t$
2. extended inverse proportional ie $I + a = k/(t + b)$
3. quadratic (modelled with three points) using the Lagrange polynomial
4. inverse cubic ie $I = a_0/t + a_1/t^2 + a_2/t^3$

The algorithms were set up using three points of data, the minimum, maximum and preferred speeds and their associated time of descents. A target time was input into each algorithm and an estimate for speed found. The time taken to descend at this estimated speed was then compared with the target time. A reasonable estimate was defined to be one within 5secs of the target time.

The inverse proportional relationship provided a rough model and gave a reasonable estimate within three iterations.

The Lagrange polynomial provided a good model, giving a reasonable estimate within two or three iterations using the low to centre preferred speeds. However, the choice of a high preferred speed had a large effect on the first estimate and up to four iterations were required.

The inverse cubic also provided a good model, giving a reasonable estimate within two or three iterations using any of the preferred speeds.

The extended inverse proportional provided the best model of the relationship between speed and time. The algorithm took the same number of iterations or less than the other approximations and was not affected unduly by the change in preferred speed. The fit of the extended inverse proportional to the actual data points was the closest of all the algorithms (Fig(12)), and had the computational advantage that its constants were smaller than those for the inverse cubic.

This algorithm has now been incorporated into the Speed Control Adviser. It has been found that the target CAS is normally given directly (to within required tolerance levels) although occasionally two or three iterations are necessary. The extended inverse proportional algorithm is therefore working effectively and keeping processing time to a minimum.

## References

[1] G Ord : *Computer Assistance in Air Traffic Systems of the 1990s*
(RSRE Memo no. 3876)

[2] SAN Magill, ACF Tyler, ET Wilkinson, W Finch :
*Computer-based tools for assisting air traffic controllers with arrivals flow management*
(RSRE REPORT No 88001)

[3] ACF Tyler : *Computer assisted aircraft arrivals management using speed control.*
COMPAS 87 conference on Computer Applications in Aircraft Design and Operation.
(Proceedings edited by TKS Murthy and JP Fielding)

[4] RA Buckingham : *Numerical Methods.* (Published by Pitman)

[5] SD Conte : *Elementary Numerical Analysis - An algorithmic approach*
(Published by McGraw Hill Book Company)

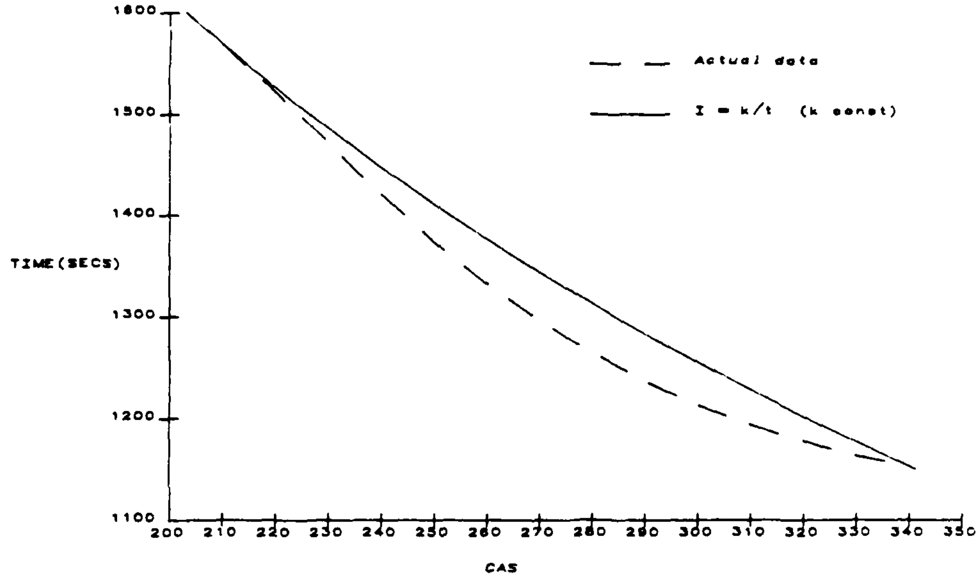


FIGURE 1

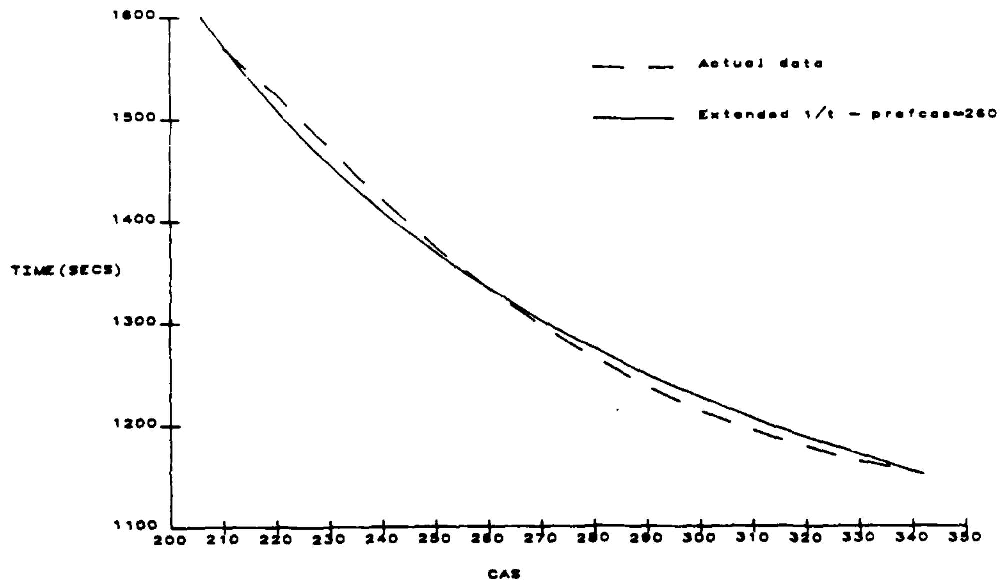


FIGURE 2

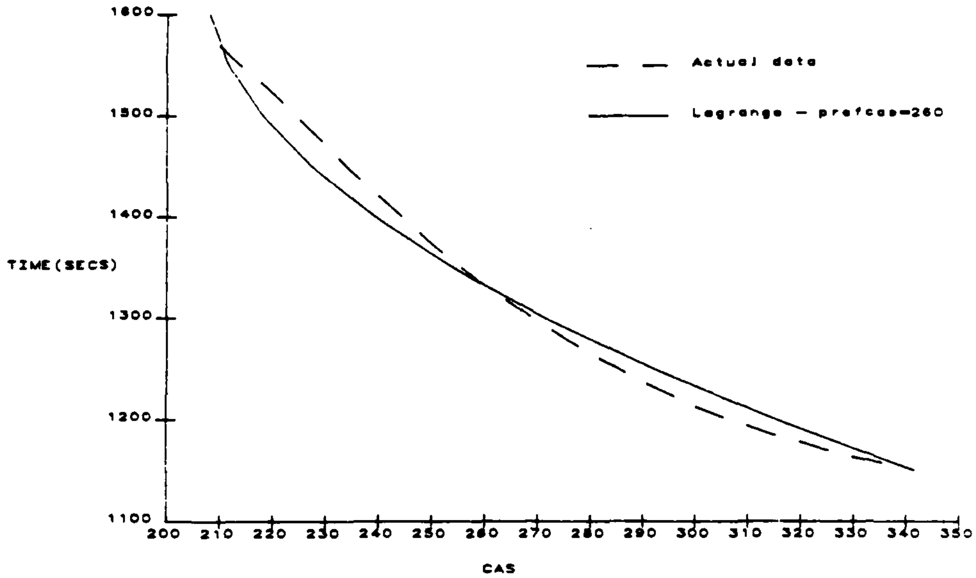


FIGURE 3

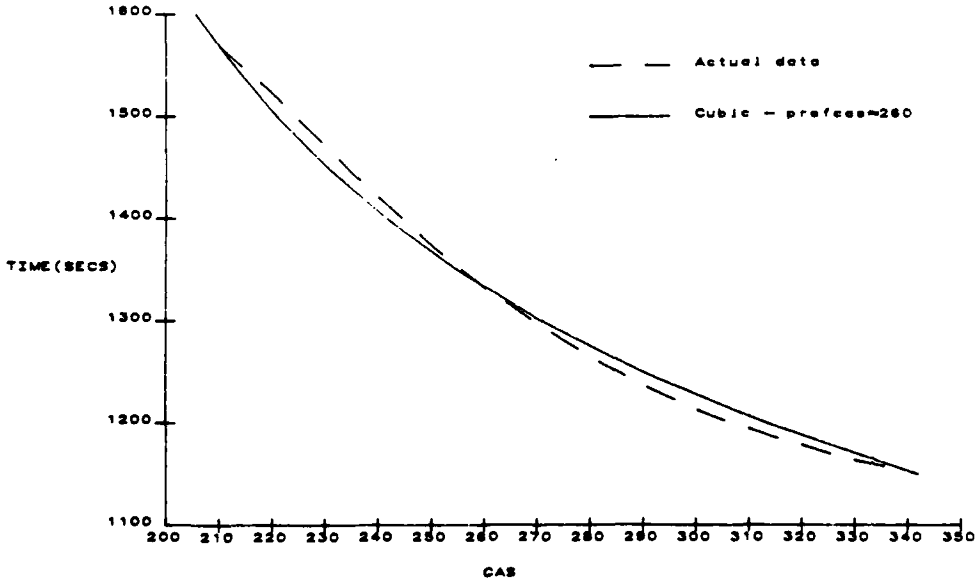


FIGURE 4

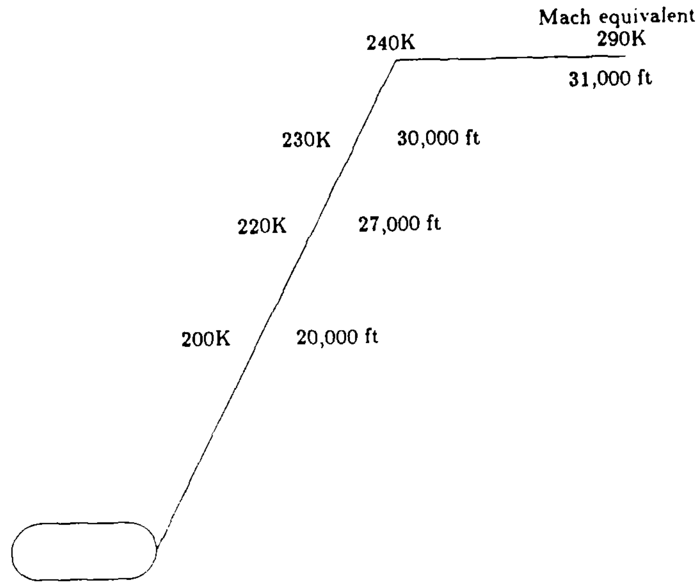


Figure 5: DESCENT PROFILE

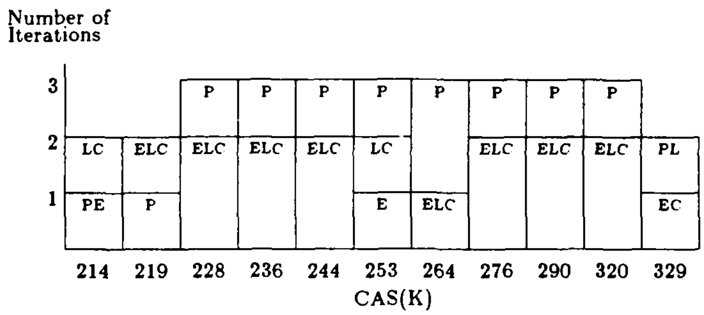


Figure 6: Preferred CAS = 260K

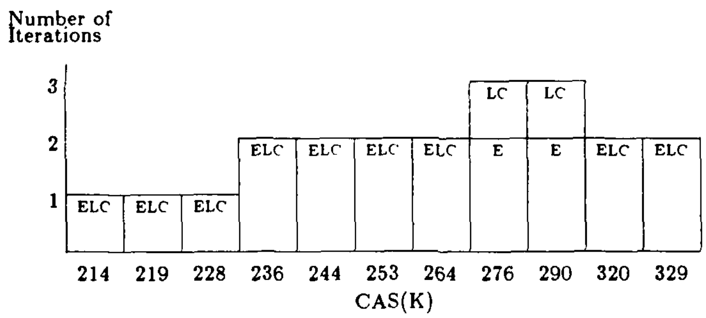


Figure 7: Preferred CAS = 230K

P = Inversely proportional L = Lagrange
E = Extended 1/T C = Inverse cubic

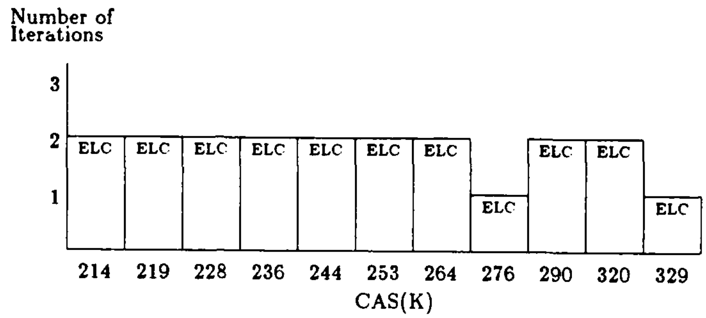


Figure 8: Preferred CAS = 275K

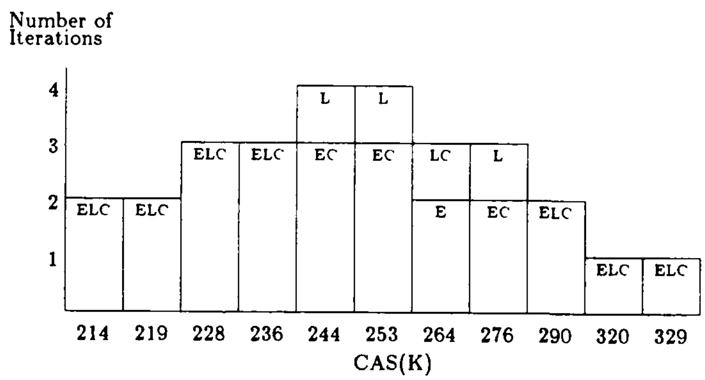


Figure 9: Preferred CAS = 320K

P = Inversely proportional  L = Lagrange
E = Extended 1/T  C = Inverse cubic

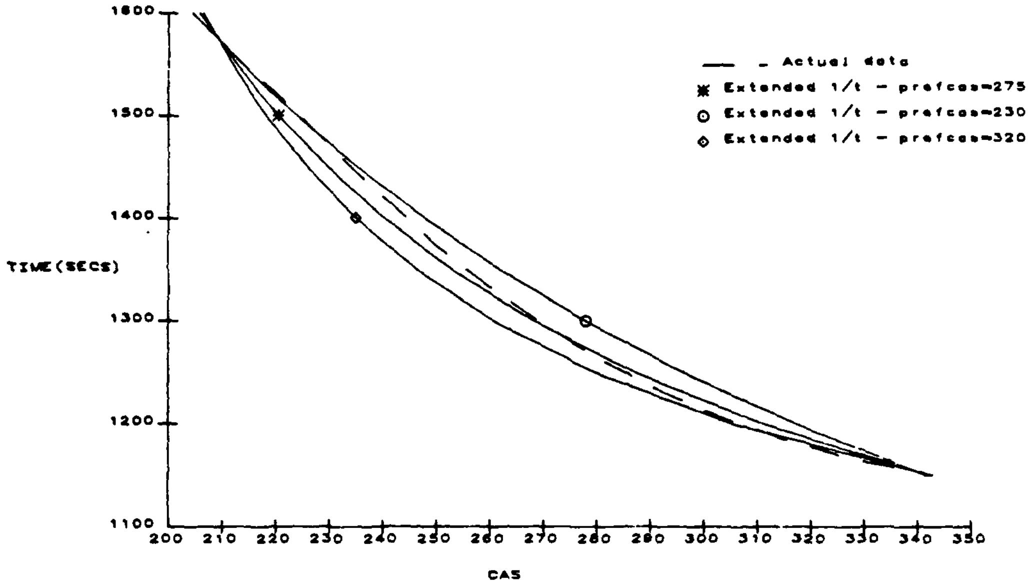


FIGURE 10

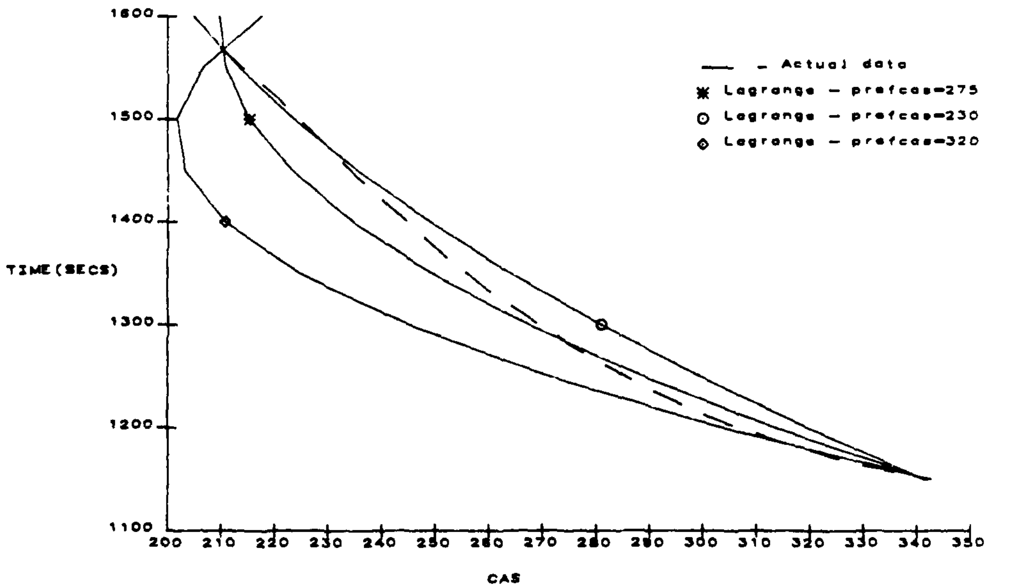


FIGURE 11

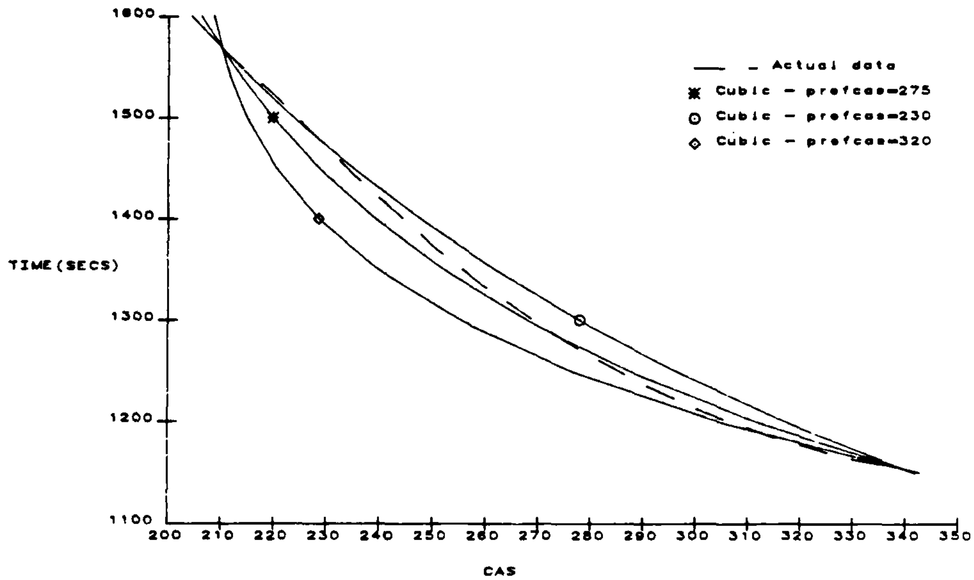
1600
1500
1400
1300
1200
1100
TIME (SECS)
200 210 220 230 240 250 260 270 280 290 300 310 320 330 340 350
CAS
_ Actual data
Cubic - prefcas=275
Cubic - prefcas=230
Cubic - prefcas=320

FIGURE 12

DOCUMENT CONTROL SHEET

Overall security classification of sheet ....UNCLASSIFIED.........................................

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification eg (R) (C) or (S) )

| 1. DRIC Reference (if known) | 2. Originator's Reference MEMO 4131 | 3. Agency Reference | 4. Report Security Classification U/C |
|---|---|---|---|
| 5. Originator's Code (if known) 7784000 | 6. Originator (Corporate Author) Name and Location ROYAL SIGNALS & RADAR ESTABLISHMENT ST ANDREWS ROAD, GREAT MALVERN, WORCESTERSHIRE WR14 3PS | | |
| 5a. Sponsoring Agency's Code (if known) | 6a. Sponsoring Agency (Contract Authority) Name and Location | | |

7. Title
COMPARISON OF INTERPOLATION ALGORITHMS FOR SPEED CONTROL IN AIR TRAFFIC MANAGEMENT

7a. Title in Foreign Language (in the case of translations)

7b. Presented at (for conference papers) Title, place and date of conference

| 8. Author 1 Surname, initials | 9(a) Author 2 | 9(b) Authors 3,4... | 10. Date | pp. ref. |
|---|---|---|---|---|
| BUDD, A J | | | 1988.08 | 19 |

| 11. Contract Number | 12. Period | 13. Project | 14. Other Reference |
|---|---|---|---|
| | | | |

15. Distribution statement
UNLIMITED

Descriptors (or keywords)

continue on separate piece of paper

Abstract
With air traffic movements at a high level, techniques to assist air traffic management using computers are being investigated. One technique in particular being studied is the early adjustment of the speed of arriving aircraft so that the rate of flow near to the airports is closely matched to landing capacity.

A Speed Control Adviser has been developed which allocates a landing time to each inbound aircraft. Once the estimated landing time is known, the speed the aircraft must fly needs to be calculated. This cannot be done directly and interpolation using a suitable polynomial approximation is used.

This memorandum investigates four polynomials and examines their effectiveness at providing a good estimate with minimum computation.

S80/48